TWIN CITY ATARI INTEREST GROUP NEWSLETTER

NOVEMBER 1983

President: Phil Seifert (448-7042) Vice President: Dale Panton
Secretary: Rick Christian Treasurer: Jim Dahlberg

OCTOBER MEETING NOTES

Phil called the meeting to order and introduced the club officers,himself the president - Phil Seifert,the V.P. - Dale Panton, the treasurer - Jim Dahlberg, and the secretary - Rick Christian.

Phil asked for more news articles for the news letter,because of late the news letter has been rather thin.

Under the exciting news category, the Atari 600XL is finally out and available. Phil brought one to the meeting to show everyone. It has the same OS Roms as the 1200XL, the cartridge slot is on top, two joystick ports on the side, and a parallel bus out put on the back. Austin-Franklin is making a 48KB Upgrade board that should be available in about 4-6 weeks.

Atari LOGO is out, Phil will hold a tutorial at next month's meeting.

Now for some (tongue in cheek) bad news - Texas Instruments Home Computer Division has been folded.

It was brought up that we again need to find a new meeting place, as we cannot use this one after the first of the year. Suggestions are being sought for a new location.

The BBS is now up and running with a few minor bugs at phone #544-9058. You can sign on and get a password by leaving a private message for the sysop, the format is illustrated in the opening flashes. There will be no fee (i.e. $5) at least until after ALL the bugs are worked out. It was brought up that maybe instead of a one time fee of $5 , the fee should be made annual. During the debate on the question, the cost of the phone line (over $40) was brought up, and Phil and Ray Voss will look into what can be done to reduce the cost. No vote was taken at this time.

From Tom Moore, we learned that our current membership stands at about 333 members.

Jim Schultz, the assistant librarian, told us that: 1. The Starwarp program is not on the cassette of the month due to the lack of room. 2. If you need an old disk of the month, see Wayne Vassal. 3. A.M.S music disks are available through the club librarian. And 4. Next month we will have a Christmas Music disk available.

Don Robinson, the Compute Magazine Program co-ordinator, said that he would be collecting disks from those that typed programs in after the meeting. If you want to type in programs from Compute, call Don at 473-3968, to avoid duplication as much as possible.

During the Q & A, it was asked if the CDC Plato cartridge was out yet- it's still in the data testing stage. The latest word on the 800XL is that it is due out before Christmas. Someone from the floor asked "Which Christmas?"

Under service tips, good service was reported from SA Electronics in St. Paul. So long for now, see you next month.

REAL TIME 3D GRAPHICS/MULTIPLICATION
by RALPH RUSSELL

Since I have owned my ATARI computer, I have spent a tremendous amount of time attempting to devise of ways to make fast 3D graphics possible. As a result I have decided that a person has to be crazy to do that. If you want to program decent 3D graphics you need a much more powerful computer. But alas, us crazy people must continue the quest. The first step in writing a 3D graphics program is to learn the mathematics involved. Perhaps at a later date I will write an article on that subject. I found a book that descibed everything I needed to know about the necessary mathematics (PRINCIPLES OF INTERACTIVE COMPUTER GRAPHICS by William M. Newman and Robert F. Sproull). After reading and studying portions of this book, and writing some BASIC programs, I determined two things. One, for what I wanted to do, the math was not to difficult and involved mostly a lot of multiplication and some division. Two, BASIC was definitely not "real time" at one image every 5 to 10 seconds.

BASIC is very helpful for testing out ideas and mathematical formulas, but I could see that if I wanted it to be fast, I would have to write it in assembly language. I was going to have work mainly on two areas. I would have come up with a line drawing routine analagous to BASIC's PLOT and DRAWTO commands. I would have to write multiplication and division routines that were much faster than the floating point routines used by BASIC. The line drawing I will postpone until a future article.

After many experiments and some more reading, I decided I would use 24 bit fixed point arithmetic. This would not be anywhere near as accurate or flexible as floating point, but it would be much faster. I would represent all of my variables as 24 bit fixed point numbers with an assumed hexadecimal point after the first 8 bits.

Ex: decimal 10.5 = hex 0A.8000
decimal 0.25 = hex 00.4000

Unfortunately, this limits us to numbers no larger than FF.FFFF, but even with this severe limitation, it is a workable situation. The 6502 processor does not have a multiply or divide instruction. It only has *add and subtract. The* standard way to multiply is the shift and add algorithm.

1. Initialize product to zero.
2. Initialize counter to 24.
3. Shift multiplier left 1 bit.
4. If carry is clear go to 6.
5. Add multiplicand to product.
6. Decrement counter: If zero, go to 9.
7. Shift product left 1 bit.
8. Go to 3.
9. Exit.

The following routine uses this algolrithm. For complete accuracy, when two 24 bit integers are multiplied, the result should be stored in a 48 bit field. But due to our decision to use 24 bit fixed point numbers, we must throw away the first 8 bits and the last 16 bits of the product so that it's alignment remains the same as the multiplier and multiplicand. This has the added benifit of cutting down on the amount of work the routine has to do. All variables are stored on page zero (low byte first).

```
SAVX =$B0           SAVE REGISTER X
SAVY =$B1           SAVE REGISTER Y
MULP =$B2           MULTIPLIER
PROD =$B6           PRODUCT
MULC =$B9           MULTIPLICAND
;
MULT  LDA #0        PRODUCT = ZERO
      STA PROD
      STA PROD+1
      STA PROD+2
      LDA MULP      EXIT IF
      ORA MULP+1    MULTIPLIER
      ORA MULP+2    = ZERO
      BNE TMULC
      RTS
TMULC LDA MULC      EXIT IF
      ORA MULC+1    MULTIPLICAND
      ORA MULC+2    = ZERO
      BNE DOIT
      RTS
DOIT  STX SAVX      SAVE INDEX
      STY SAVY      REGISTERS
      LDA #0        FULL PRODUCT
      STA PROD-1    = ZERO
      LDX #24       COUNTER = 24
      BNE ASLM      GO TO SHIFT MULP
SHIFT ASL A         SHIFT PRODUCT
      ROL PROD-1    LEFT 1 BIT
      ROL PROD
      ROL PROD+1
      ROL PROD+2
ASLM  ASL MULP      SHIFT MULTIPLIER
      ROL MULP+1    LEFT 1 BIT.
      ROL MULP+2
      BCC CHCNT     NO ADD IF CARRY
                    IS CLEAR.
```

```
        CLC             ADD MULTIPLICAND
        ADC MULC        TO PRODUCT
        TAY
        LDA MULC+1
        ADC PROD-1
        STA PROD-1
        LDA MULC+2
        ADC PROD
        STA PROD
        TYA
        BCC CHCNT
        INC PROD+1
        BNE CHCNT
        INC PROD+2
CHCNT   DEX             COUNTER=COUNTER-1
        BNE SHIFT       DONE IF ZERO
        LDX SAVX        RESTORE INDEX
        LDY SAVY        REGISTERS
        RTS
```

This routine is many times faster than the BASIC floating point multiply, but not fast enough.

To achieve even faster execution speed, I was going to have to use lookup tables. The concept is simple. Instead of using the shift and add algorithm, I would use a 4 bit by 4 bit lookup table. This table uses 256 bytes of memory and can be set up using .BYTE statements.

```
MTBL =*
    .BYTE 0,0,0,0,0,0,0,0
    .BYTE 0,0,0,0,0,0,0,0
    .BYTE $00,$01,$02,$03,$04,$05,$06,$07
    .BYTE $08,$09,$0A,$0B,$0C,$0D,$0E,$0F
    .BYTE $00,$02,$04,$06,$08,$0A,$0C,$0E
    .BYTE $10,$12,$14,$16,$18,$1A,$1C,$1E
    .BYTE $00,$03,$06,$09,$0C,$0F,$12,$15
    .BYTE $18,$1B,$1E,$21,$24,$27,$2A,$2D
    .BYTE $00,$04,$08,$0C,$10,$14,$18,$1C
    .BYTE $20,$24,$28,$2C,$30,$34,$38,$3C
    .BYTE $00,$05,$0A,$0F,$14,$19,$1E,$23
    .BYTE $28,$2D,$32,$37,$3C,$41,$46,$4B
    .BYTE $00,$06,$0C,$12,$18,$1E,$24,$2A
    .BYTE $30,$36,$3C,$42,$48,$4E,$54,$5A
    .BYTE $00,$07,$0E,$15,$1C,$23,$2A,$31
    .BYTE $38,$3F,$46,$4D,$54,$5B,$62,$69
    .BYTE $00,$08,$10,$18,$20,$28,$30,$38
    .BYTE $40,$48,$50,$58,$60,$68,$70,$78
    .BYTE $00,$09,$12,$1B,$24,$2D,$36,$3F
    .BYTE $48,$51,$5A,$63,$6C,$75,$7E,$87
    .BYTE $00,$0A,$14,$1E,$28,$32,$3C,$46
    .BYTE $50,$5A,$64,$6E,$78,$82,$8C,$96
    .BYTE $00,$0B,$16,$21,$2C,$37,$42,$4D
    .BYTE $58,$63,$6E,$79,$84,$8F,$9A,$A5
    .BYTE $00,$0C,$18,$24,$30,$3C,$48,$54
    .BYTE $60,$6C,$78,$84,$90,$9C,$A8,$B4
    .BYTE $00,$0D,$1A,$27,$34,$41,$4E,$5B
    .BYTE $68,$75,$82,$8F,$9C,$A9,$B6,$C3
    .BYTE $00,$0E,$1C,$2A,$38,$46,$54,$62
    .BYTE $70,$7E,$8C,$9A,$A8,$B6,$C4,$D2
    .BYTE $00,$0F,$1E,$2D,$3C,$4B,$5A,$69
    .BYTE $78,$87,$96,$A5,$B4,$C3,$D2,$E1
```

To demonstrate the use of this table, suppose we want to multiply 8 and 7. The following assembly instructions are all that is necessary.

```
LDA #8
ASL A
ASL A
ASL A
ASL A
ORA #7          RESULT OF OR = $87
TAX
LDA MTBL,X      LOOKUP PRODUCT
```

After execution, the accumulator will contain $38 (decimal 56) the product of 8 x 7. It now becomes apparent that to multiply larger numbers, we're going to have to do a lot of LSR and ASL instructions. So I decided I would use two more tables to replace the shift instructions. Instead of coding these tables with byte statements, I wrote a very simple routine to do it.

```
SHTR = $8000
SHTL = $9000

        LDX #0
SHFL    TXA
        LSR A
        LSR A
        LSR A
        LSR A
        STA SHTR,X
        TXA
        ASL A
        ASL A
        ASL A
        ASL A
        STA SHTL,X
        DEX
        BNE SHFL
```

The following substitutions can now ne made.

```
LDA ADDR
LSR A       IS          LDX ADDR
LSR A   REPLACED        LDA SHTR,X
LSR A       BY
LSR A

LDA ADDR
ASL A       IS          LDX ADDR
ASL A   REPLACED        LDA SHTL,X
ASL A       BY
ASL A
```

Multiplication by using this method has two advantages. One, it is faster because it executes fewer instructions. Two, it is faster because it allows you to skip large chunks of code if a given nybble in the multiplier = zero. It also has one disadvantage. It uses a large amount of memory (768 bytes just for tables). The 24 bit routine itself is quite large because it is written in straight line code (no looping), and it has to lookup 36 partial products and add them up.

This is where I really get crazy. The 4 bit by 4 bit lookup table was working great and my graphics were faster, but I was not satisfied. I decided to try a 4 bit by 8 bit lookup table for even greater speed. The table would require 8192 bytes and would be located at $8000 and $9000. There was no way I was going to code that many .BYTE statements, so I wrote a routine to build the tables. Because the result of a 4 bit by 8 bit multiplication could be as large as 12 bits, I actually have to have two tables, one for the low byte of the product ($8000) and one for the high byte ($9000). Here is the routine to build the tables. It is very dificult to follow, but it works.

```
LOPR  = ANYWHERE      Product low
HIPR  = ANYWHERE      Product high
TABLO = $8000         Product table
TABHI = $9000         Product table

BMTAB =*
      LDA #0          Init low
      STA $80         pointers to
      STA $82         product tables
      TAY
MTL1  LDA #[TABLO/256]+1
      STA $81
      LDA #[TABHI/256]+1
      STA $83
      TYA
      STA LOPR
      LDA #0
      STA HIPR
MTL2  LDA LOPR
      STA ($80),Y
      LDA HIPR
      STA ($82),Y
      INC $81
      INC $83
      LDA $81
      CMP #[TABLO/256]+16
      BEQ NXTY
      TYA
      CLC
      ADC LOPR
      STA LOPR
      BCC MTL2
      INC HIPR
      JMP MTL2
NXTY  DEY
      BNE MTL1
```

Now that we have built our table, multiplication becomes much simpler. Because we will be testing the multiplier for zero, we can use the first page of each table for something else. We will still need the two shift tables so we will store them at $8000 and $9000. Whenever we lookup a partial product using the high order nybble of a byte in the multiplier, it must be shifted left 4 bits for proper alignment before being added to the final product. The following routine will multiply two 8 bit numbers and give a 16 bit result.

```
SAVX = $B0          SAVE X REGISTER
SAVY = $B1          SAVE Y REGISTER
MULP = $B2          MULTIPLIER
MULC = $B3          MULTIPLICAND
PROD = $B4          PRODUCT (2 BYTES)

MULT  LDA #0        PRODUCT = ZERO
      STA PROD
      STA PROD+1
      STX SAVX      SAVE REGISTERS
      STY SAVY

      LDY MULC
      LDA MULP
      CLC
      TAX
      AND #$0F      GET LOW NYBBLE
      BEQ MTPL2     TEST FOR ZERO

      ORA #TABLO/256  SET UP PAGE
      STA $81         ZERO POINTERS
      ORA #TABHI/256
      STA $83

      LDA ($80),Y   GET LOW PRODUCT
      STA PROD
      LDA ($82),Y
      STA PROD+1

MTPL2 LDA SHTR,X    GET HIGH NYBBLE
      BEQ DONE      TEST FOR ZERO

      ORA #TABLO/256  SET UP PAGE
      STA $81         ZERO POINTERS
      ORA #TABHI/256
      STA $83
```

```
        LDA ($80),Y  GET HIGH PRODUCT
        TAX          SHIFT AND ADD
        LDA SHTL,X   TO PRODUCT
        ADC PROD
        STA PROD
        LDA ($82),Y
        TAY
        LDA SHTL,Y
        ADC SHTR,X
        ADC PROD+1
        STA PROD+1

DONE    LDX SAVX     RESTORE REGISTERS
        LDY SAVY
        RTS
```

Note: It is assumed that $80 and $82 both contain zero.

The 24 bit fixed point routine is more complex but follows the same logic. The partial products are stored in two seperate areas, one for those partial pruducts that do not need to be shifted (PROD), and one for those that do (PRD2). After all 6 nybbles of the multiplier have been processed, PROD and PRD2 are added (with appropriate shifts). The following is the 24 bit routine I am currently using in my graphics programs.

```
SAVX = $B0
SAVY = $B1
MULP = $B2     MULTIPLIER
PROD = $B7     PRODUCT
MULC = $BA     MULTIPLICAND
PRD2 = $BF     SHIFTED PRODUCT

SHTR = $8000  SHIFT TABLE (RIGHT)
SHTL = $9000  SHIFT TABLE (LEFT)
TABLO = $8000 LOOKUP TABLE (LOW)
TABHI = $9000 LOOKUP TABLE (HIGH)

MULT LDA #0
     STA PROD
     STA PROD+1
     STA PROD+2
     LDA MULP
     ORA MULP+1
     ORA MULP+2
     BNE TMLC
     RTS

TMLC LDA MULC
     ORA MULC+1
     ORA MULC+2
     BNE DOIT
     RTS

DOIT LDA #0
     STA PROD-2
     STA PROD-1
     STA PRD2-2
     STA PRD2-1
     STA PRD2
     STA PRD2+1
     STA PRD2+2
     STX SAVX
     STY SAVY

MTPL1 LDA MULP
     CLC
     TAX
     AND #$0F
     BEQ MTPL2
     ORA #TABLO/256
     STA $81
     ORA #TABHI/256
     STA $83

     LDY MULC
     LDA ($80),Y
     STA PROD-2
     LDA ($82),Y
     LDY MULC+1
     ADC ($80),Y
     STA PROD-1
     LDA ($82),Y
     LDY MULC+2
     ADC ($80),Y
     STA PROD
     LDA ($82),Y
     ADC #0
     STA PROD+1

MTPL2 LDA SHTR,X
     BEQ MTPL3
     ORA #TABLO/256
     STA $81
     ORA #TABHI/256
     STA $83

     LDY MULC
     LDA ($80),Y
     STA PRD2-2
     LDA ($82),Y
     LDY MULC+1
     ADC ($80),Y
     STA PRD2-1
     LDA ($82),Y
     LDY MULC+2
     ADC ($80),Y
     STA PRD2
     LDA ($82),Y
     ADC #0
     STA PRD2+1

MTPL3 LDA MULP+1
     TAX
     AND #$0F
     BEQ MTPL4
     ORA #TABLO/256
     STA $81
     ORA #TABHI/256
     STA $83

     LDY MULC
     LDA ($80),Y
     ADC PROD-1
     STA PROD-1
     LDA ($82),Y
     LDY MULC+1
     ADC ($80),Y
     BCC ADPR1
     INC PROD+1
     CLC
ADPR1 ADC PROD
     STA PROD
     LDA ($82),Y
     LDY MULC+2
     ADC PROD+1
     ADC ($80),Y
     STA PROD+1
     LDA ($82),Y
     ADC #0
     STA PROD+2

MTPL4 LDA SHTR,X
     BEQ MTPL5
     ORA #TABLO/256
     STA $81
     ORA #TABHI/256
     STA $83

     LDY MULC
     LDA ($80),Y
     ADC PRD2-1
     STA PRD2-1
     LDA ($82),Y
     LDY MULC+1
     ADC ($80),Y
     BCC ADPR2
     INC PRD2+1
     CLC
ADPR2 ADC PRD2
     STA PRD2
     LDA ($82),Y
     LDY MULC+2
     ADC PRD2+1
     ADC ($80),Y
     STA PRD2+1
     LDA ($82),Y
     ADC #0
     STA PRD2+2
```

```
MTPL5 LDA MULP+2
      TAX
      AND #$0F
      BEQ MTPL6
      ORA #TABLO/256
      STA $81
      ORA #TABHI/256
      STA $83

      LDY MULC
      LDA ($80),Y
      ADC PROD
      STA PROD
      LDA ($82),Y
      LDY MULC+1
      ADC ($80),Y
      BCC ADPR3
      INC PROD+2
      CLC
ADPR3 ADC PROD+1
      STA PROD+1
      LDA ($82),Y
      LDY MULC+2
      ADC PROD+2
      ADC ($80),Y
      STA PROD+2
      CLC

MTPL6 LDA SHTR,X
      BEQ SUMM
      ORA #TABLO/256
      STA $81
      ORA #TABHI/256
      STA $83

      LDY MULC
      LDA ($80),Y
      ADC PRD2
      STA PRD2
      LDA ($82),Y
      LDY MULC+1
      ADC ($80),Y
      BCC ADPR4
      INC PRD2+2
      CLC
ADPR4 ADC PRD2+1
      STA PRD2+1
      LDA ($82),Y
      LDY MULC+2
      ADC PRD2+2
      ADC ($80),Y
      STA PRD2+2

SUMM  CLC
      LDX PRD2-2
      LDA SHTL,X
      ADC PROD-2
      LDY PRD2-1
      LDA SHTL,Y
      ORA SHTR,X
      ADC PROD-1
      STA PROD-1
      LDX PRD2
      LDA SHTL,X
      ORA SHTR,Y
      ADC PROD
      STA PROD
      LDY PRD2+1
      LDA SHTL,Y
      ORA SHTR,X
      ADC PROD+1
      STA PROD+1
      LDX PRD2+2
      LDA SHTL,X
      ORA SHTR,Y
      ADC PROD+2
      STA PROD+2

      LDA PROD-1    ROUND UP IF
      BPL MDNE      NECESSARY
      INC PROD
      BNE MDNE
      INC PROD+1
      BNE MDNE
      INC PROD+2
MDNE  LDX SAVX
      LDY SAVY
      RTS
```

There are many useful degenerate cases of this routine. This version assumes that only the low nybble of the middle byte of the multiplier might be non zero (MULP = $00.0x00).

```
SMLT LDA #0
     STA PROD
     STA PROD+1
     STA PROD+2
     LDA MULP+1
     BNE TMLCS
     RTS
TMLCS LDA MULC
     ORA MULC+1
     ORA MULC+2
     BNE MTPL1S
     RTS

MTPL1S LDA MULP+1
     CLC
     ORA #TABLO/256
     STA $81
     ORA #TABHI/256
     STA $83

     LDY MULC
     LDA ($82),Y
     LDY MULC+1
     ADC ($80),Y
     STA PROD
```

```
LDA ($82),Y
LDY MULC+2
ADC ($80),Y
STA PROD+1
LDA ($82),Y
ADC #0
STA PROD+2
RTS
```

Note: The Y register may need to be saved.

Another extremely fast version assumes that you are multiplying by a 4 bit constant. I use this routine, right in line (no JSR), to multiply a 24 bit variable by a 4 bit integer constant. This example assumes the multiplier is 3 ($03.0000), and that no overflow will occur. Execution times are: 40 cycles if page zero varibles are used, and 46 cycles if not.

```
CLC
LDX MULC
LDA TABLO+$300,X
STA PROD
LDA TABHI+$300,X
LDX MULC+1
ADC TABLO+$300,X
STA PROD+1
LDA TABHI+$300,X
LDX MULC+2
ADC TABLO+$300,X
STA PROD+2
```

None of the multiplication routines described in this article handle signs. The sign of the product must be handled by the calling routine. Just remember the following rule: If the signs of the multiplier and multiplicand are opposite, the sign of the product is negative, if the signs are the same, the product is positive. Another problem is that none of the 24 bit routines can handle overflow. The calling routine must insure that the values used for the multiplier and multiplicand will not generate a product that doesn't fit into 24 bits.

As I have been writing, another possible way to speed up the 24 bit routine has occured to me. If we set up two additional 4096 byte tables at $A000 and $B000 which contain the same data as those located at $8000 and $9000, shifted to the left 4 bits, we can eliminate the need for building two seperate partial products and summing them at the end. Instead, because all partial products would be aligned when we look them up, we can summ all of the partial products as we generate them. This could save us about 80 cycles or so for each non zero multiply.

If I get ambitious I will write some articles in the future that could include the following topics.

1. Assembly language line drawing.
2. How to code division routines and then speed them up.
3. The basic mathematics of 3D graphics.

I have donated a 3D simulation program to the club library which demonstrates many of the routines in action. If anyone would like a copy of this program, I will gladly make you one for $1, if you furnish me with a disk. Also if anyone has any questions about the material presented here, give me a call (after 6) at 881-5952.

16 BIT MUSIC
by RALPH RUSSELL

I have owned my ATARI 800 since January of 1982. I have been a member of TAIG since February of 1982. During all of this time I have never written an article for the newsletter. On several occasions I have promissed that I would write one, but because of my obsession with programming, I never took the time to do so.

Typically, Atari computers are programmed to play music using four seperate voices. This is accomplished by setting four sets of hardware registers with 8 bit values that control the pitch and intensity of the music. As the Atari hardware manual points out, it is possible to combine these registers in such a way to produce much better pitch. The main problem with producing music in this manner is there are only two voices instead of four. However there are many fine pieces of music that can be played using only two voices. I have written such a program. It is written in assembly language and plays TWO PART INVENTION IN D MINOR by J. S. BACH. I have donated this program (object and source code) to the club library. I hope it will show up on the disk of the month one day.

The first step to utilizing the 16 bit music capability of your computer is to set the appropriate registers. The following assembly instructions will initialize your computer into the 16 bit mode.

```
LDA #0        INITIALIZE POKEY
STA $D208
LDA #3
STA $232
STA $D20F
```

```
LDA #$78      CLOCK CHANNELS 1 & 3 WITH
STA $D208     1.79 MHz.
              JOIN CHANNELS 3 & 4.
              JOIN CHANNELS 1 & 2.

LDA #$A0      SET VOICES TO PURE
STA $D201     (MUSICAL) TONES AND
STA $D203     VOLUME OFF
STA $D205
STA $D207
```

Now that we are in the 16 bit mode all we have to do now is plug the proper pitch values into the frequency control registers.

$D200 = LOW BYTE OF VOICE #1
$D202 = HI BYTE OF VOICE #1
$D204 = LOW BYTE OF VOICE #2
$D206 = HI BYTE OF VOICE #2

The volume of each voice is controled by placing the proper value in the audio control registers.

$D203 = VOLUME CONTROL FOR VOICE #1
  #$A0 = VOICE OFF
  #$AF = LOUDEST VOLUME
$D207 = VOLUME CONTROL FOR VOICE #2
  #$A0 = VOICE OFF
  #$AF = LOUDEST VOLUME

Note: The high order nybble of these registers must be $A or we generate 16 bit noise and not music.

The next step is to determine the exact frequencies of all of the notes we want to play. To accomplish this, I went to the local library and found a book about electronic systhesizers. This book just happened to contain the complete 9 octave equally tempered chromatic scale frequencies (in Hertz) in a nice little table.

Next we must calculate the 16 bit values necessary to control the frequency registers. The formula is provided by the Atari hardware manual.

$$Fout = \frac{Fin}{2(audf+7)}$$

Fout = Frequency output
Fin = Frequency input (1789790 Hz)
audf = 16 bit register value

As you can see, we must solve this equation for audf to get what we need.

$$audf = \frac{1{,}789{,}790}{2(Fout)} - 7$$

The following is a table of the 16 bit values necessary to control the frequency registers, calculated using the above formula.

PITCH	FREQ	audf(DEC)	audf(HEX)
C	16.352	54720	D5C0
C#	17.324	51649	C9C1
D	18.354	48750	BE6E
D#	19.445	46015	B3BF
E	20.601	43432	A9A8
F	21.826	40994	A022
F#	23.124	38693	9725
G	24.499	36521	8EA9
G#	25.956	34470	86A6
A	27.500	32535	7F17
A#	29.135	30708	77F4
B	30.867	28985	7139
C	30.703	27357	6ADD
C#	34.648	25821	64DD
D	36.708	24372	5F34
D#	38.890	23004	59DC
E	41.203	21712	54D0
F	43.653	20493	500D
F#	46.249	19342	4B8E
G	48.999	18527	4751
G#	51.913	17231	434F
A	55.000	16264	3F88
A#	58.270	15351	3BF7
B	61.735	14489	3899
C	65.406	13675	356B
C#	69.295	12907	326B
D	73.416	12182	2F96
D#	77.781	11498	2CEA
E	82.406	10853	2A65
F	87.307	10243	2803
F#	92.499	9668	25C4
G	97.998	9125	23A5
G#	103.82	8613	21A5
A	110.00	8128	1FC0
A#	116.54	7672	1DF8
B	123.47	7241	1C49
C	130.81	6834	1AB2
C#	138.59	6450	1932
D	146.83	6088	17C8
D#	155.56	5746	1672
E	164.81	5423	152F
F	174.61	5118	13FE
F#	184.99	4831	12DF
G	195.99	4559	11CF
G#	207.65	4303	10CF
A	220.00	4061	0FDD
A#	233.08	3832	0EF8
B	246.94	3617	0E21

C	261.63	3413	0D55
C#	277.18	3222	0C96
D	293.66	3040	0BE0
D#	311.13	2869	0B35
E	329.63	2708	0A94
F	349.23	2555	09FB
F#	369.99	2412	096C
G	391.99	2276	08E4
G#	415.31	2148	0864
A	440.00	2027	07EB
A#	466.16	1913	0779
B	493.88	1805	070D
C	523.25	1703	06A7
C#	554.37	1607	0647
D	587.33	1517	05ED
D#	622.25	1431	0597
E	659.26	1350	0546
F	698.46	1274	04FA
F#	739.99	1202	04B2
G	783.99	1134	046E
G#	830.61	1070	042E
A	880.00	1010	03F2
A#	932.32	953	03B9
B	987.77	899	0383
C	1046.50	848	0350
C#	1108.73	800	0320
D	1174.66	755	02F3
D#	1244.51	712	02C8
E	1318.51	672	02A0
F	1396.91	634	027A
F#	1479.98	598	0256
G	1567.98	564	0234
G#	1661.22	532	0214
A	1760.00	501	01F5
A#	1864.66	473	01D9
B	1975.53	446	01BE
C	2093.00	421	01A5
C#	2217.46	397	018D
D	2349.32	374	0176
D#	2489.02	353	0161
E	2637.02	332	014C
F	2793.83	313	0139
F#	2959.96	295	0127
G	3135.97	278	0116
G#	3322.44	262	0106
A	3520.00	247	00F7
A#	3729.31	233	00E9
B	3951.07	219	00DB
C	4186.01	207	00CF
C#	4434.92	195	00C3
D	4698.64	183	00B7
D#	4978.03	173	00AD
E	5274.04	163	00A3
F	5587.66	153	0099
F#	5919.92	144	0090
G	6271.93	136	0088
G#	6644.88	128	0080
A	7040.00	120	0078
A#	7458.63	113	0071
B	7902.13	106	006A

Like the 8 bit music, as the pitch gets higher, the interval between each note becomes shorter. This means that the highest octave might not be as accurate as it should be. This is not really a problem because these are extremely high notes that probably won't be used anyway (Only dogs can hear them I think). Because of the 8 bit nature of the 6502 processor, it will make programming music easier if we store our audf(H) values in two seperate tables. The first table (FREQL) is the low byte of the register value and the second table (FREQH) is the high byte. If, for example, you want voice #1 to play an A (440 Hz), The following assembly code will do the job.

```
LDX #57          A(440) IS THE 58th
                 NOTE IN THE TABLE.
LDA FREQL,X      GET LOW BYTE.
STA $D200        STORE LOW BYTE.
LDA FREQH,X      GET HIGH BYTE.
STA $D202        STORE HIGH BYTE.
LDA ##A8         SET VOLUME HALF WAY
STA $D203        UP FOR VOICE #1.
```

The following are the frequency tables as they appear in the program I wrote.

```
FREQL = *
    .BYTE $C0,$C1,$6E,$BF,$A8,$22
    .BYTE $25,$A9,$A6,$17,$F4,$39
    .BYTE $DD,$DD,$34,$DC,$D0,$0D
    .BYTE $8E,$51,$4F,$88,$F7,$99
    .BYTE $6B,$6B,$96,$EA,$65,$03
    .BYTE $C4,$A5,$A5,$C0,$F8,$49
    .BYTE $B2,$32,$C8,$72,$2F,$FE
    .BYTE $DF,$CF,$CF,$DD,$F8,$21
    .BYTE $55,$96,$E0,$35,$94,$FB
    .BYTE $6C,$E4,$64,$EB,$79,$0D
    .BYTE $A7,$47,$ED,$97,$46,$FA
    .BYTE $B2,$6E,$2E,$F2,$B9,$83
    .BYTE $50,$20,$F3,$C8,$A0,$7A
    .BYTE $56,$34,$14,$F5,$D9,$BE
    .BYTE $A5,$8D,$76,$61,$4C,$39
    .BYTE $27,$16,$06,$F7,$E9,$DB
    .BYTE $CF,$C3,$B7,$AD,$A3,$99
    .BYTE $90,$88,$80,$78,$71,$6A
FREQH = *
    .BYTE $D5,$C9,$BE,$B3,$A9,$A0
    .BYTE $97,$8E,$86,$7F,$77,$71
    .BYTE $6A,$64,$5F,$59,$54,$50
    .BYTE $4B,$47,$43,$3F,$3B,$38
    .BYTE $35,$32,$2F,$2C,$2A,$28
    .BYTE $25,$23,$21,$1F,$1D,$1C
    .BYTE $1A,$19,$17,$16,$15,$13
    .BYTE $12,$11,$10,$0F,$0E,$0E
    .BYTE $0D,$0C,$0B,$0B,$0A,$09
```

```
.BYTE $09,$08,$08,$07,$07,$07
.BYTE $06,$06,$05,$05,$05,$04
.BYTE $04,$04,$04,$03,$03,$03
.BYTE $03,$03,$02,$02,$02,$02
.BYTE $02,$02,$02,$01,$01,$01
.BYTE $01,$01,$01,$01,$01,$01
.BYTE $01,$01,$01,$00,$00,$00
.BYTE $00,$00,$00,$00,$00,$00
.BYTE $00,$00,$00,$00,$00,$00
```

TAIG HALL OF FAME-MDN

Sorry I missed the last couple of meetings but I had to go up north. Since I missed the last meeting, I will be taking high scores over the phone. Just dial (612) 869-2077 and ask for Mark. Remember to bring your scores to the meeting and put them in the box. Here are some of my high scores and don't forget to call!!

1. MDN-DigDug-114,800
2. MDN-Jawbreaker-55,180
3. MDN-Donkey Kong-122,000
4. MDN-Crisis Mountain-38,800
5. Don Nelson-Apple Panic-153,930

TAIG Tips
by MDN

Since I missed the last meeting I will be taking questions or answers over the phone. My number is (612) 869-2077. I have not been playing many adventures lately but have some questions on some. Don't forget to bring your questions and answers to the meeting and put them in the box. Here are some questions you adventurers might want to answer. (No laughing allowed!!!!)

1. How do I get out of the desert in Sands of Egypt!!! (Quit Laughing!!)
2. Zork 2-How do I get past that &^!!*+/ lizard door.
3. Deadline-How do I find the secret closet.

NOTE...There are maps for all you Ultima]I[lovers. Maps of all the planets, all the cities, all the towns, and all the villages. The book also tells all the trasactions in all the cities etc. It also tells how to win. To get one call (612) 869-2077 and ask for Mark. The books will be from $2.00 to $3.00 dollars. Thanks, MDN

HARDWARE/SOFTWARE NOTES
By Todd Burkey

A lot of new things this month. First, RANA is really shipping their drives now with double density, single density, and 1050 extended singled density format capacities. These drives are truly intelligent in that they perform continuous error checking on their own hardware as well as being able to let you format and do read/write verification of disks from the drives' front panel. The first few that I got in had flaky ROMS (they weren't burned in long enough according to RANA), but the replacements and the other new ones I got in look pretty good. Of all the drives on the market right now for the ATARI, I like the RANA the best...it is small, inexpensive, and local repair centers are being designated (according to the local RANA rep). Now for the bad news. RANA's local rep is starting to give COMPUTER FOOD and my company (SOFT UNLIMITED) a hard time because we aren't store fronts. He told me and Steve that he won't commit any deliveries to us, since several store fronts have complained about our pricing (i.e. since we had our orders placed so long ago, we had received a lower introductory price on the units than the stores can get them for now.) He also had the audacity to say that people without store fronts don't support their customers and/or show them how to use the equipment. Hogwash. I think I provide even more support than most stores do because I don't sell something just to make a profit, I do it also to get more involved in the computer community and learn even more. That is probably the only reason I have been in business longer than any of the other local computer stores dealing in ATARI. Most store fronts are more motivated by trends and chances for a quick profit, than worrying about whether the product they are selling really fits the customers needs. This is examplified with what happened on the RANA'S. I have had a number of people call me up and tell me that several of the stores were telling them to buy PERCOMS or 1050'S because the RANA's were 'too new', 'un-reliable', 'incompatible', and 'piles of junk'. All of these were likely substantiated more on the fact that the stores had PERCOMS or 1050s in stock rather than on any technical experience on the store owners' part. Now that the RANA's are looking good, these stores are so interested in stocking the RANAs that they apparently want the market all to themselves. Welcome to the world of business. So why do I still

support the RANA? Because it has more potential than any drive right now, and I will stock them even if I have to go to the president of RANA with this problem. Enough said. Also, please note that ON LINE COMPUTER CENTERS was NOT, I repeat NOT, one of the stores that I referred to above. As far as I know, they were the only storefront to make any major orders with RANA early on in the game.

What about all of the other drives on the market today? The Percom drives are still around and new versions are coming out with printer ports, slimline drives, etc. The drives themselves are pretty reliable, being based on TANDON bare drives with a PERCOM controller. The TRAK drives are a possible alternative to the RANA if you have a monitor and don't need 1050 compatiblility. You need a monitor because other clubs' newsletters indicated that the drives had a lot of RF problems and this was verified by another dealer that has one in stock in the cities. Otherwise the drives are roughly the same size and cost (although the TRAK drives really are from a newer company and may suffer from repair delays). The ASTRA drives are purportedly being shipped, although I haven't seen one yet and I have read in other newsletters that they have a problem with blowing up their power supplies. The 1050 drive does not support double density, regardless of what the stores have been telling you! I get a phone call a day asking about this. The Micro-Mainframe drive I will believe when I see it delivered. The only other drive that comes to mind is a new one that just came out. I can't recall the name off-hand, but you will recognize it when you see it. Apparently it comes with software for word processing, spreadsheet work, and other applications. Seems a little overkill, but I will know more next month (late flash-Steve from COMPUTER FOOD just ordered one of these drives, so might have it by the next meeting.)

Now it is time to contradict myself somewhat. If you are thinking about getting a pair of drives for your ATARI and maybe later a printer and/or modem, then I don't recommend that you buy any of the above mentioned drives...RANA included. In that case I suggest you consider a configuration using the ATR8000 and bare drives. Part of the reason is shown in the chart below:

ITEM	LOCAL COST	RETAIL COST
ATR8000 (16K) -	$315	$400
2 SSDD DRIVES -	$410	$500
TOTAL COST -	$725	$900
2 RANA DRIVES -	$630	$900
1 850 INTER. -	$160	$220
TOTAL COST -	$780	$1120

The rest of the reason is that for an additional $100 you can get CP/M and 64K on the ATR8000 and actually have the ability to run KAYPRO, OSBORNE, XEROX, and another of other computers' software. Plus you will have a second computer and drives that you can put on most any other computer you decide to buy in the future. Also, for a little more money, you can add 8 inch drives to your system and give yourself around 2.5 Million Bytes of storage with a single pair of drives (that equates to over 30 810 drives).

Enough of disks and on to something more fun....all of the games that came out over the last month or so.
First, in the area of twitch games (play them long enough and you will twitch in your sleep...ever have weird dreams after playing space invaders?) I give very high marks to OIL'S WELL. This is a game similiar to PAC MAN in that you have to clear dots off of the screen, but that is where the similarity ends. In this game, you have to tunnel your way through the underground passages chomping up dots and the little monsters as well. You lay a retractable pipe behind you (which you have to protect from the monsters) and work through increasingly complex mazes. This game is very fast paced and short in duration. A second twitch game was B.C.'s Quest For Tires. This one I recommend soley on its graphics animation. Straight from the HART comic strip, you get to ride a tire through obstacles, over turtle backs on a river, past the Fat Broad with a club, up a hill with rocks flying past you, over a pit with the help of the bird (the one that usually carries the turtle around), speed down the hill and jump a ravine, over a river past a dinosaur, through a cavern with more obstacles, and finally to the rescue of the Cute Chick. A nice game, but it only took me about 15 trys before I made it through the first round and got to the Chick.

In the area of adventures, the sequel to ALI BABA is now out. The new game is called Heracles (not Hercules as I originally thought) and is much improved over ALI BABA. Not only is the music and text much better, but the game play is improved and the mythological theme was well researched and the player has some well defined goals to achieve. I recommend this game to anyone who is tired of typing away at the text adventures and likes to just sit back and have full control with a joystick. Also out is ULTIMA III. This adventure is similar to ULTIMA II in the area of graphics, but the music has been improved and the game play isn't as viscious. It also supports multiple players. For those of you who have to have every adventure that comes out, also out is BENEATH APPLE MANOR, Planetfall, and Enchanter.

Finally, In the area of languages, BASICXL and LOGO are out. Both look like superb packages which I hope someone will be kind enough to review for the next newsletter.

CP/M ON ATARI
By Todd Burkey

Well, we have enough ATR8000 owners now to start a small club, so I am taking the initiative and calling the first meeting on December 17 at 3:00 in the afternoon. The purpose of this meeting is to make sure everyone is up to speed on their systems and there will be a general swap of all of the public domain software that everyone has gotten in the past several months. If we have time, some demos of how to run certain utilities in CP/M will also be performed.

This first meeting will be limited to those members who have CP/M systems (both attached and unattached to their ATARI'S) or those of you who truly plan on upgrading your ATARI to run CP/M. If you are interested in attending, please give me a call at 542-1027.

If you already have a CP/M system and an ATARI computer (two seperate systems), then I will have something in next months' newsletter you will be able to use. Ralph Jenson found some info on a BBS that describes how to allow your ATARI to access your CP/M systems' disk drives and ports (printer and modem) through the ATARI serial bus. This is done in somewhat the same manner as the ATR does internally, although you won't be accessing CP/M from your ATARI keyboard.

ATARI MUSIC
by Mike Davis

Let's get into entering some music. Below are two lines of our mystery tune. So boot up your A.M.S. program, and let's take another look at the main menu.

First we must remove the A.M.S. disk and put in a disk for us to save our music on. If the disk isn't formatted, just press "I" then "Y" to have it formatted.

Now back at the main menu we press "B". Notice that since we don't have any music entered yet, voice 1 is automatically selected and we're at measure 1. I usually use voice 1 for the melody which is usually the highest note. Before we enter any notes, let's get our meter and key set. On our song you'll notice a "Cents" sign. This is 4/4 time cut. Rather than explain that, just type "MET(return)", and then type "4/4(return)". Now type "KEY(return)". Looking at the music next to the treble clef is one sharp sign. So type "1#(return)". You do not have to enter which notes are sharp or flat when entering the key. Just enter the number of sharps or flats.

From this point on, I shall use an "*" to represent hitting the return key. Starting with the top notes of the first measure, enter:

```
RQ*
B4S*
D*
F*
A*
FT*
A*
FQ..*
```

Now we shall discuss the entries. A.M.S. has default values for octave, duration, and volume. These values will change when a new value is used. For example, when RQ is entered the duration on the right of the screen will indicate a "Q" for a quarter time. Our next entry (B4S) causes the duration to change to "S" for a sixteenth time. The next entry (D) caused a D5S/mf to be entered due to the defaults. The duration was "S", and the volume was "mf". Also one other factor now comes to light. The scale option was ON. When the note that you want to enter is within 3 or 4 notes of the previous entry, then you can just enter the letter because the octave will be automatically added and changed if

necessary. With Scale OFF, the "D" that was entered would have ended up being a D4 instead of a D5. This helps to simplify note entry at times.

Continuing down the list, we come to F5T/mf and A5T/mf which are 32nd notes. I realize 16th notes are shown, but when I played back the measure, using "P*", those two notes didn't sound like a trill or short grace notes as designated. So I shortened them to 32nd notes. That's all fine and well, but we still have to get our measure to have exactly 4 beats. An F5H/mf would have been too long. So I calculated that a double-dotted quarter would fit just right to make exactly 4 beats.

Before you get discouraged, please realize I picked this piece of music for its difficulty in entry. Most music is much more straight forward.
Make the following entries:

```
M* (This gets you into measure 2)
REPM1* (This will repeat measure 1's notes)
M* (Onto measure 3)
RQ*
C5S*
E*
G*
B*
GT*
BT*
GQ..*
M* (Onto measure 4)
BQ*
G*
E*
B4*
M* (Onto measure 5)
Q* (Let's "Quit" here)
```

Now we have the first four measures for voice 1 completed. Now let's listen to voice 1 by pressing "A", which is play. Another menu is now on the screen. Since all we have done is voice 1, press "L" and listen to what we have so far. When measures are completed, the program drops back to the main menu. The tempo can be changed by typing "H" and then entering a new number.

To start entering voice 2 notes, type "B" then "2". The meter and key should still be the same so we can get on with note entries. We are now in measure 1 for voice 2. Typically, I use the next to the highest note for voice 2, but sometimes other decisions must be taken into consideration. For example, effectively we have six notes to be played at one time during this measure, but we only have four voices. Decision time is now upon us. After the quarter rest, B4S, F4H, D4H, B3H, A3H, and the rest of G2H. should be played. B4S is already being handled by voice 1. Normally you would not repeat notes that are the same but on different octaves. In this case, B3H could be tossed out, but the remaining notes don't repeat. You would now just choose one or experiment by playing the completed measure (all four voices together) to determine which note combination sounded the best.

Starting with measure 1 of voice 2 enter:

```
RQ*
F4H*
RQ*
M* (Onto measure 2)
RQ*
A4H*
RQ*
M* (Onto measure 3)
RQ*
G4H*
RQ*
M* (Onto measure 4)
E5H*
CQ*
F4*
Q* (Let's "Quit" here)
```

Let's listen to both voice 1 and 2 together. Press "A" then "F". By the way, you can also listen to particular measures by typing a "D". You will then be prompted for what measure or series of measures. To demonstrate, type "D" then "2,4(return)". Now you're at the second menu requesting what voices. Press "F". You are now listening to voices 1 and 2 play measures 2 through 4. This is a very good feature when you only want to listen to a few measures that are well into the song.

Onto voice 3. Press "B" then "3". Enter the following:

```
RQ*
D4H*
RQ*
M* (Onto measure 2)
RQ*
FH*
RQ*
M* (Onto measure 3)
RQ*
EH*
RQ*
```

```
M* (Onto measure 4)
C5H*
AQ*
EF*
Q* (Let's "Quit" here)
```

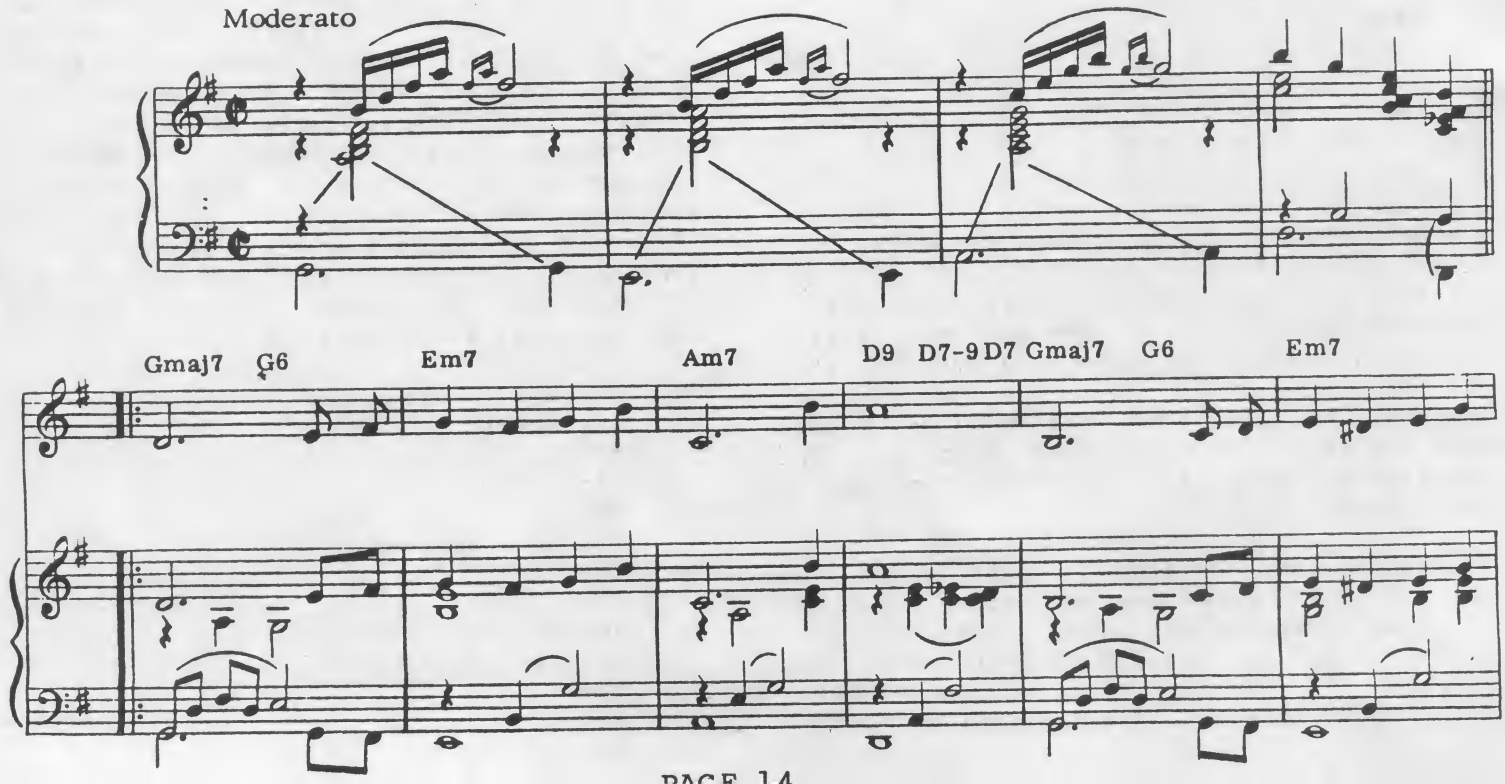
Onto voice 4. Press "B" then "4". Enter the following:

```
G2Q*
A3H*
G2Q*
M* (Onto measure 2)
E2Q*
D4H*
E2Q*
M* (Onto measure 3)
A2Q*
A3H*
A2Q*
M* (Onto measure 4)
D3Q*
G3H*
FQ*
Q* (Let's "Quit" here)
```

Voice 4 entry might be confusing some of you due to the way I had you enter the notes. The reason for not entering a dotted-half and then a quarter of the same bass note is due to how boring that would sound on A.M.S. Bass or accompliment movement is best. For example in measure 1, I shortened G2 to a quarter to match the time of the quarter rest above it. Then I chose A3H to complete the chord. Next came G2Q to end the measure. By having voice 4 make these jumps, it gives the bass more movement. In this case, I had some latitude in note selection, but many times, unless you improvise, this may not be possible due to the arrangement that you are entering.

Let's listen to all four voices. Press "A" then "A" again. Now save this work by typing "E" and the name, for example "SONG.AMS". It is not necessary to put "D:" in front the file name. To look at the disk directory, type "G". You should see "SONG.AMS" in the listing. Just press the (return) key to get back to the main menu. Note that when you came back your song is still in memory even though there is no "MEM.SAV" on the disk. "MEM.SAV" is not necessary since you're not actually loading all of DOS.

You have just completed the introduction of the song. The next measure, 5, is the beginning of the melody that I'm sure most people will recognize. Notice that measures 5 through 10 have three staffs. You can ignore the top treble cleff staff. It is just the melody only which is already contained in the staff below it. Next month, I shall list the entries for those six measures.



ULTIMA II

by Mike Davis

Are you tired of getting killed off? Or are you disgusted with having to kill everything in sight in hopes of getting a helm, skull key, blue tassles, and etc.? Take heart for this article could be just what you need.

First let me state that I realize that true adventurers will consider the release of this information as nothing short of treason. But likewise for those of us who don't necessarily enjoy spending hundreds of hours in game play, this should add to our enjoyment of the adventure since I shall not be giving away any clues or hints.

This article concerns itself with disk editting of the "Player Disk". This is NOT the "MASTER PLAY DISK", but the player disk which you generate in accordance with the documentation. Sectors 16 and 17 (decimal) contain your character's attributes, weapons, armour, spells, and other possessions.

Let's start with sector #16. I used "Diskedit" (version 2.0 by Todd Burkey of Soft Unlimited) to examine and modify the sectors. Either version 1.0 or 2.0 will work in the following explanation. First insert your Basic cartridge and boot up "Diskedit". When you get the main menu, insert your player disk into the disk drive. Type "16(return)".

Sector 16 is now displayed on your screen. Notice that the bytes are arranged in 16 rows with 8 bytes per row. The number to the left of each row is the hexidecimal location of the first byte of that row. Remember hexidecimal counts 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, and etc. Below is a byte location table of sector 16.

Sector 16 (Decimal)

Byte #	Attribute	Comments
10	Sex	CD=Male C6=Female
11	Profession	00=Fighter 01=Cleric 02=Wizard 03=Thief
12	Race	00=Human 01=Elf 02=Dwarf 03=Hobbit
15	Strength	Decimal #
16	Agility	"
17	Stamina	"
18	Charisma	"
19	Wisdom	"
1A	Intelligence	"
1B	Hit Points	High #
1C	Hit Points	Low #
1D	Food	High #
1E	Food	Low #
20	Experience	High #
21	Experience	Low #
22	Gold	High #
23	Gold	Low #
2B	Weapon	In hand 00=Hand 01=Dagger 02=Mace 03=Axe 04=Bow 05=Sword 06=Great Sword 07=Light Sword 08=Phaser 09=Quick Sword
2C	Armour	Wearing 00=Skin 01=Cloth 02=Leather 03=Chain 04=Plate 05=Reflect 06=Power
2D	Spell	Ready to Use 01=Light 02=Down Ladder 03=Up Ladder 04=Passwall 05=Surface 06=Prayer 07=Magic Missile 08=Blink 09=Kill
2E	Torches	How Many?
2F	Keys	" "
30	Tools	" "
41	Daggers	How Many?
42	Maces	" "
43	Axes	" "
44	Bows	" "
45	Swords	" "
46	Great Swords	" "
47	Light Swords	" "
48	Phasers	" "
49	Quick Swords	" "

62	Cloth	"	"
63	Leather	"	"
64	Chain	"	"
65	Plate	"	"
66	Reflect	"	"
67	Power	"	"

Let's discuss sector 16's locations and data within those locations. Byte 10 should have either "CD" or "C6" in that location. Bytes 11 and 12 should have one of the four data bytes shown. Bytes 15 through 1A are decimal numbers representing the amount of each of the six characteristics. Bytes 1B through 23 are also decimal numbers for hit points, food, experience, and gold, but these bytes are read a little different. The "High #" digits are thousands and hundreds. While the "Low #" digits are tens and units. For example, if hit points "High #" was 75, and the "Low #" was 50. Then your hit point total would be 7550. Bytes 2B through 2D are what weapon, armour, or spell that you have ready to use. Bytes 2E through 30, 41 through 49, and 62 through 67 are read as decimal numbers. One point to keep in mind concerns rollover. In other words if you give yourself 99 keys and you happen to get one more through a battle or some other transaction, you will end up with 00 keys. This is a possibility with any quantity data.

In order to edit a byte within a sector, press "E". You should notice the cursor blinking in the upper left corner of the screen at byte location 00. Using the "Ctrl" key and an "arrow" key, you can move the cursor to any byte that you wish to change. For example, let's give ourselves a strength of 95. Move the cursor to byte location #15 (hexidecimal). Now enter "$95(Return)". You now have a "95" in that location on the screen, but you don't have it on the disk yet. Press the "Esc" key. Now press "W". You are now in the write mode of "Diskedit". It is prompting you for which sector the screen information is to be written. Notice that the default value is already 16. So just press the "Return" key to write the updated data onto your player disk. One of the important points is not to forget to write your editted sector back onto your player disk. Your player disk is now ready to use with the Ultima II program.

Below is a table for sector 17. It contains "How Many" (decimal) of what spells or items.

Sector 17 (Decimal)

Byte # (Hex)	Description	Comments
01	Lights	How Many?
02	Down Ladders	" "
03	Up Ladders	" "
04	Passwalls	" "
05	Surfaces	" "
06	Prayers	" "
07	Magic Missiles	" "
08	Blinks	" "
09	Kills	" "
20	Rings	" "
21	Wands	" "
22	Staffs	" "
23	Boots	" "
24	Cloaks	" "
25	Helms	" "
26	Gems	" "
27	Ankhs	" "
28	Red Gems	" "
29	Skull Keys	" "
2A	Green Gems	" "
2B	Brass Buttons	" "
2C	Blue Tassles	" "
2D	Strange Coins	" "
2E	Green Idol	" "
2F	Try-Lithiums	" "

This concludes the information that I have on the Ultima II player disk. As I stated earlier, this isn't for everybody, but I feel for many of us it will make the adventure much more enjoyable.

Atari Music Notes
by Mike Davis

The response for Christmas music contributions was very well received. Dan Lokken, Steve Crowley, Mike Buchman, Ellen Moore, Rick Dempsey, Mike Gustafuson, Don Robinson, and Phil (the Ruthless) Seifert contributed many songs. Because of this, we now have a Christmas disk with both sides filled with a wide variety of music.

All contributions were greatly appreciated. One in particular that I enjoyed very much was from Mike Buchman. He had an arrangement of "Santa Claus is Coming to Town" that I thought was superb. Its file name on the disk is "SANTA.AMS".
name on the disk is "SANTA.AMS".

I wish to thank Ellen Moore for editting and preparing side two of the disk. Without her help, I probably would not have been able to complete the disk by our November meeting.

I would like to remind everyone that we are always looking for more music in any category. Volume 2 is almost complete. Side 2 is over half-filled. So I figure volume 2 to be completed by the January meeting.

There will be 20 copies of the Christmas disk at the next meeting. Because they are double-sided copies, the price will be $6.00 and can be purchased from the club library.

From The Editor
By Todd Burkey

Wow, some good sized articles again. I was getting worried that the newsletter was going to shrink to nothing by years' end. If I have room when I paste this issue up, I will insert the instructions on uploading articles for the newsletter. I will also be amienable to downloading articles anyone wants to upload to the TAIG BBS. Just give me a call whenever something is uploaded (I prefer this method to doing an attended ATARI to ATARI transfer).

Special Request: Does anyone want to take care of the club's 400 computer. All you have to do is keep it at home and be sure to bring it to every meeting (for the demos). Contact any of the board members if you are interested.

Well, I won't be at the meeting this month, since I'll be going to the COMDEX show in Las Vegas. I hear that there are going to be a lot of new computers introduced at significantly reduced prices. I hope ATARI has some surprises in store for us. So, I'll see you in December and let you know what's going on.(If you find you have a desperate need for diskettes before the December meeting, give me a call at home.)

Spokesmen for a local electronics firm have announced a computer program that virtually eliminates lost time due to malfunctions of computer components, called "OREMA" (oh-ray-ma) from the latin-oremus, meaning "let us pray." the program offers prayers at selected time intervals for the continued in tegerity of memory units, tape transports, and other elements subject to depravity.

Basically liturgical in nature and structure, OREMA uses standard petitions and intercessions stored on magnetic tapes in latin, hebrew and fortran. It holds regular maintenance services thrice daily on an automatic cycle, and operator intervention is required only for mounting tapes and making responses such as "and with thy spirit" , on the console typewritter.

Prayers in hebrew and fortran are offered directly to the central processor and disc storage units, but latin prayers may go to other peripheral equipment for transfer to the central processor by internal subroutines.

Although manufacturer-supplied prayer reels cover all machine troubles known today, the program will add card prayers to to any tape, as needed, after the final existing amen block. Classified prayer reels are available for goverment installations.

In trials on selected machines, OREMA reduced, by 92.3% , the average down time due to component failure. The manufacturer's spokesman emphasized however, that OREMA presently defends only against hardware malfunctions. Requestor errors and other human blunders, will continue unchecked until completion of a future version ; to be called sin-OREMA.

PRAIRIE MICROSYSTEMS, INC.

Software Development / Computer Sales – Services

941-4504

ATARI

ODDS AND ENDS SALE

THIS MONTH WE ARE OFFERING SUBSTANTIAL SAVINGS ON THE FOLLOWING ITEMS. MANY ARE ON SALE AT THE LOWEST PRICE EVER FOR PRAIRIE, AND MAYBE EVEN THE LOWEST PRICE EVER SEEN LOCALLY! QUANTITIES ARE LIMITED TO STOCK ON HAND (AS INDICATED) AND ALL ITEMS ARE NEW EXCEPT FOR ONE DEMO MONITOR.

HARDWARE	RETAIL	SALE
OUR ROCK-BOTTOM LOWEST PRICE ON THE INCOMPARABLE... PROWRITER 8510 PARALLEL PRINTERS (3)	$495	$360!!
IN-HOME B-KEY KEYBOARD FOR THE 400 (1)	$ 69.95	$ 49
ANCHOR SIGNALMAN MARK II MODEM (WITH CABLE) (1)	$ 99	$ 75
BMC 12" GREEN SCREEN MONITORS (2)		
ONE DEMO	$109	$ 70
ONE NEW IN BOX	"	$ 80
BOOKS		
KIDS & THE ATARI (1)	$ 19.95	$ 13
THE BOOK OF ATARI SOFTWARE 1983 (4)	$ 19.95	$ 6

SOFTWARE

WE STILL HAVE A FEW PIECES OF SOFTWARE LEFT ON CLEARANCE, PRICED AT $10 EACH OR 2 FOR $17, WITH THE EXCEPTION OF THE HOME ACCOUNTANT (2 COPIES) WHICH IS PRICED AT $49. ALL ARE ON DISK EXCEPT AS SHOWN OTHERWISE.

ENTERTAINMENT: CHOPLIFTER, CANYON CLIMBER, BANDITS, CROSSFIRE + MORE

EDUCATION: KINDERCOMP, TEACHER'S PET, EDUFUN, INVITATION TO PROGRAMMING 1 (CASSETTE)

HOME MANAGEMENT: HOME ACCOUNTANT, HOME FINANCIAL MANAGEMENT (CASSETTE)

AT PRICES LIKE THESE, HOW CAN YOU RESIST??

---

* P R A I R I E M I C R O S Y S T E M S * 9 4 1 - 4 5 0 4 *

8711 Black Maple Dr. Eden Prairie, MN 55344

SOFT UNLIMITED CHRISTMAS SPECIALS

The following prices are in effect for the month of December. Some quantities are limited, so don't wait for the meeting.

HARDWARE

64K ATR8000	$415. This will allow you to run CP/M software and use standard bare drives from your ATARI.
Tandon Drives	$225. These are the bare drives for the ATR or can be used as add on drives for the PERCOMs. Power supply and case included.
RANA Drives	$320. Finally out and being shipped. Single, Double, and 1050 capability.

OTHER

Diskettes	$18/box. Pick from elephant, wabash, or generic brands.
Flip Files	Holds 50-80 diskettes.
Software	Both ATARI and CP/M software at mail order prices.

CALL 542-1027 Evenings For Ordering Information

TWIN CITY ATARI INTEREST GROUP

MEMBERSHIP APPLICATION

Name: _____ Date: _____

Family members to be included: _____

_____

_____

Street: _____

City: _____ State: _____ ZIP: _____

Phone: (_____)_____

Membership dues are $10/yr (single or family). Make your check payable to the Twin City ATARI Interest Group (TAIG).

Dues paid: $____(check/cash) Date: _____ Received by: _____

Return to: TAIG
Tom & Ellen Moore, Membership Chairpersons
4648 31st Ave. So.
Mpls., MN 55406

TWINCITY ATARI INTEREST GROUP
6824 QUEEN AVE. S.
RICHFIELD, MN 55423

+800-538-8543

Next TAIG Meeting:
Sunday, November 27, 1983
Tutorials (org.) - 6:00
TAIG - 7:00 P.M.
COMPUTE! Planning- afterwards

Odd Fellow's Lodge
3003 27th Ave. S.
Mpls, MN